Claims

1. A method of producing a combinational circuit, the method comprising:

determining cyclic parameter; and

synthesizing a combinational circuit in accordance with the determined cyclic parameters.

2. A method as defined in Claim 1, wherein determining cyclic parameters further comprises:

defining at least one input variable;

defining at least one output variable; and

defining a relationship between the at least one input and at least one output variable whereby the relationship includes a cycle.

3. A method as defined in Claim 2, further including at least one internal variable.

4. A method as defined in Claim 2, wherein the relationship includes structured dependency between an input and output variable.

5. A method as defined in Claim 1, wherein determining cyclic parameters further comprises:

defining at least one input variable;

defining at least one internal variable; and

defining a relationship between the at least one input and at least one internal variable whereby the relationship includes a cycle.

6. A method as defined in Claim 5, wherein at least one internal variable is an output variable.

7. A method as defined in Claim 5, wherein the relationship includes structured dependency between an input and internal variable.

8. A method as defined in Claim 1, wherein the cyclic parameters are used in a logic synthesis process.

9. A method as defined in Claim 8, wherein the cyclic parameters are used in a structuring operation of the logic synthesis process.

10. A method as defined in Claim 9, wherein the structuring operation includes a substitution phase of the logic synthesis process.

11. A method as defined in Claim 1, wherein the method of producing a combinational circuit is optimized.

12. A method as defined in Claim 11, wherein the combinational circuit is optimized with regard to cost.

13. A method as defined in Claim 12, wherein the cost is measured as an area.

14. A method as defined in Claim 13, wherein the area is determined by a literal count.

15. A method as defined in Claim 13, wherein the area is determined by a gate count.

16. A method as defined in Claim 11, wherein the combinational circuit is optimized with regard to performance.

17. A method as defined in Claim 16, wherein the performance is measured as a delay of the combinational circuit.

18. A method as defined in Claim 11, wherein the combinational circuit is optimized with regard to fault tolerance.

19. A method as defined in Claim 11, wherein the combinational circuit is optimized with regard to power consumption.

20. A method as defined in Claim 11, wherein the combinational circuit is optimized with regard to testability.

21. A method as defined in Claim 1, wherein synthesizing comprises:

creating a network with no cycles;

introducing a cycle into the network;

determining if the network is combinational; and

repeating introduction of cycles into the network until a desired combinational circuit is implemented.

22. A method as defined in Claim 21, wherein the combinational circuit is optimized.

23. A method as defined in Claim 22, wherein the combinational circuit is optimized with regard to cost.

24. A method as defined in Claim 23, wherein the cost is measured as an area.

25. A method as defined in Claim 24, wherein the area is determined by a literal count.

26. A method as defined in Claim 24, wherein the area is determined by a gate count.

27. A method as defined in Claim 22, wherein the combinational circuit is optimized with regard to performance.

28. A method as defined in Claim 27, wherein the performance is measured as a delay of the combinational circuit.

29. A method as defined in Claim 22, wherein the combinational circuit is optimized with regard to fault tolerance.

30. A method as defined in Claim 22, wherein the combinational circuit is optimized with regard to power consumption.

31. A method as defined in Claim 22, wherein the combinational circuit is optimized with regard to testability.

32. The method as defined in Claim 21, wherein introducing, determining and repeating is performed in a structuring operation of logic synthesis.

33. A method as defined in claim 32, wherein the structuring operation includes a substitution phase of logic synthesis.

34. A method as defined in Claim 1, wherein synthesizing comprises:

creating a densely interconnected network;

excluding edges from the network;

determining if the network is combinational; and

repeating excluding edges the network until a desired combinational circuit is implemented.

35. A method as defined in Claim 34 wherein the combinational circuit is optimized.

36. A method as defined in Claim 35, wherein the combinational circuit is optimized with regard to cost.

37. A method as defined in Claim 36, wherein the cost is measured as an area.

38. A method as defined in Claim 37, wherein the area is determined by a literal count.

39. A method as defined in Claim 37, wherein the area is determined by a gate count.

40. A method as defined in Claim 35, wherein the combinational circuit is optimized with regard to performance.

41. A method as defined in Claim 40, wherein the performance is measured as a delay of the combinational circuit.

42. A method as defined in Claim 35, wherein the combinational circuit is optimized with regard to fault tolerance.

43. A method as defined in Claim 35, wherein the combinational circuit is optimized with regard to power consumption.

44. A method as defined in Claim 35, wherein the combinational circuit is optimized with regard to testability.

45. The method as defined in Claim 34, wherein introducing, determining and repeating is performed in a structuring operation of logic synthesis.

46. A method as defined in claim 45, wherein the structuring operation includes a substitution phase of logic synthesis.

47. A method of logic synthesis, the method comprising:

determining cyclic parameters; and

using the cyclic parameters during synthesis of a combinational circuit.

48. A method as defined in Claim 47, wherein the cyclic parameters are used in a structuring operation of logic synthesis.

49. A method as defined in Claim 48, wherein the structuring operation includes a substitution phase of the logic synthesis.

50. A logic synthesizer comprising:

a set of cyclic parameters; and

a processor configured to synthesize a combinational circuit in accordance with the determined cyclic parameters.

51. A synthesizer as defined in Claim 50, wherein the cyclic parameters further comprises:

at least one input variable;

at least one output variable; and

a relationship between the at least one input and at least one output variable whereby the relationship includes a cycle.

52. A synthesizer as defined in Claim 51, further including at least one internal variable.

53. A synthesizer as defined in Claim 51, wherein the relationship includes structured dependency between an input and output variable.

54. A synthesizer as defined in Claim 50, wherein the cyclic parameters further comprises:

at least one input variable;

at least one internal variable; and

a relationship between the at least one input and at least one internal variable whereby the relationship includes a cycle.

55. A synthesizer as defined in Claim 54, wherein at least one internal variable is an output variable.

56. A synthesizer as defined in Claim 54, wherein the relationship includes structured dependency between an input and internal variable.

57. A method as defined in Claim 50, wherein the cyclic parameters are used in a logic synthesis process.

58. A synthesizer as defined in Claim 57, wherein the cyclic parameters are used in a structuring operation of the logic synthesis process.

59. A synthesizer as defined in Claim 58, wherein the structuring operation includes a substitution phase of the logic synthesis process.

60. A synthesizer as defined in Claim 50, wherein the synthesized combinational circuit is optimized.

61. A synthesizer as defined in Claim 60, wherein the combinational circuit is optimized with regard to cost.

62. A synthesizer as defined in Claim 61, wherein the cost is measured as an area.

63. A synthesizer as defined in Claim 62, wherein the area is determined by a literal count.

64. A synthesizer as defined in Claim 62, wherein the area is determined by a gate count.

65. A synthesizer as defined in Claim 60, wherein the combinational circuit is optimized with regard to performance.

66. A synthesizer as defined in Claim 65, wherein the performance is measured as a delay of the combinational circuit.

67. A synthesizer as defined in Claim 60, wherein the combinational circuit is optimized with regard to fault tolerance.

68. A synthesizer as defined in Claim 60, wherein the combinational circuit is optimized with regard to power consumption.

69. A synthesizer as defined in Claim 60, wherein the combinational circuit is optimized with regard to testability.

70. A synthesizer as defined in Claim 50, wherein the processor creates a network with no cycles, and then introduces cycles into the network and determines if the network is combinational, and then it repeats introducing cycles into the network until a desired combinational circuit, is implemented.

71. A synthesizer as defined in Claim 50, wherein the processor creates a densely interconnected network, and then excludes edges from the network and determines if the network is combinational, and then repeats excluding edges from the network until a desired combinational circuit is implemented.